many of these programs running for the growing community of collaborative users, we have implemented orderly scheme of introducing the new versions slowly. Old versions are removed only when there are no longer any SYSOUT's on them. At the same time we actively encourage users to keep their programs up-to-date to minimize the maintenance problems with LISP versions no longer supported.

TENEX-SAIL is maintained by Dr. Robert Smith of IMSSS/SUMEX and is exported through SUMEX so excellent service is locally available. A PRINT feature was added to SAIL this year in response to user suggestions as well as a number of new runtime routines. This year the RECORD data structure became available in SAIL and an effort has been made to familiarize users with this structure which is well-suited to AI applications. A collection of utilities for SAIL programming has also been gathered at SUMEX and introduced to the SAIL users.

A very important new part of the SAIL system is the BAIL interactive debugger which was written by Mr. John Reiser of SU-AI but in close consultation with SUMEX/IMSSS using the TENEX facilities to test the BAIL/TENEX interaction. BAIL allows users to interactively examine and change the contents of previously-defined variables and to enter SAIL statements using a subset of the language. Mr. Reiser offered a BAIL class at Stanford to introduce his system to local users.

LISP Efficiency:

There has been an on-going debate this past year between advocates of INTERLISP and ILISP over the relative efficiencies of the two languages and the level of assistance the language systems provide the user in developing programs. These issues are important because they influence the time required to develop new AI programs and subsequently the incremental load placed on the SUMEX machine when in use. A number of people have contributed to an evaluation of these two LISP's including Dr. R. Smith (IMSSS), Dr. Tom Wolpert (formerly of IMSSS), Mr. Larry Masinter (Xerox PARC), and Mr. Larry Fagan (MYCIN project - formerly USC-ISI). The tests were based on an implementation of a subset of REDUCE (a symbolic algebra manipulator). The results of several iterations in program refinement by experts in the respective languages were that the runtimes for the two versions were quite comparable (far less than the 5-10 disparity predicted by ILISP enthusiasts). A more disquieting result was the substantial difference in runtimes depending on how particular functions were coded IN THE SAME LANGUAGE. It is apparent from the results that factors of 10 differences in time can result from a superficial implementation - expert programming insight is essential to efficient program performance. This is not a real surprise in that it is true of programming in any language - the problems may be increased by such a rich language as INTERLISP with such a wide array of ways to do the same thing but with little guidance as to the relative costs. It has proven very difficult to quantify the "rules" for good programming. Mr. Masinter and Mr. Phil Jackson attempted to document good INTERLISP programming habits and issued a bulletin for SUMEX users.

A further impact of these data is that it is very difficult to simultaneously develop a new AI program and make the implementation highly efficient. With the iterations required to develop the conceptual design of the program, it is difficult to ensure its efficiency. This may lead to the need to reimplement the program after the basic development stabilizes to increase efficiency while still accommodating convenient and orderly further development. Such reimplementation may or may not by best done in LISP - this will depend on many factors including the nature of the program data structure requirements and anticipated further development efforts.

II.A.2.f MAINSAIL OVERVIEW

Another aspect of SUMEX's role in encouraging community software sharing which has received substantial attention this past year is the set of problems involved in software exportation. The following is a general description of the on-going development of a machine-independent programming system (MAINSAIL) by Mr. Clark Wilcox of our staff. A more detailed description of the language elements can be found in Appendix E on page 208.

The MAINSAIL programming system (referred to as SAILEX in the 1975 annual report) has undergone extensive development during the past year. The considerable interest expressed to date from across the country indicates that MAINSAIL could be a significant step towards the distribution of portable software (programs which can be executed, without alteration, on a variety of computer systems). In response, SUMEX is pursuing plans to make MAINSAIL available to other sites, and to promote the exchange of programs within a diverse computer community. This type of language and program sharing is now made more difficult by the incompatabilities among the various implementations of current languages. MAINSAIL embodies a unified approach which presents to the user the same programming system, regardless of what computer or operating system supports its execution.

SUMEX, in its role as a nationally shared computer resource, is an appropriate vehicle for the development of high-quality software unbound by the underlying machine environment. We have a built-in community of program developers acutely aware of the significance of providing their work to a broader base of users. This intersection of hardware capability, software expertise, and dedication to resource sharing presents a unique opportunity to promote a system designed for program sharing.

MAINSAIL is being developed for two closely-related reasons: as a general-purpose programming system, and as a tool for research into the design of a machine-independent programming system. MAINSAIL will be fully implemented and actively used on a number of machines. It is perhaps one of the most highly-developed languages available for the mini-

computer environment. Its machine-independent design allows it to be used for the development of programs on one computer which will be executed on another. This capability will be of increasing importance as smaller dedicated computers are used in conjunction with larger general-purpose computers, and as programs are more readily exchanged over computer networks.

The MAINSAIL language is derived from SAIL, a programming language developed at Stanford University's Artificial Intelligence Laboratory. It is not compatible with SAIL, since SAIL was designed for a PDP-10 with TOPS-10, and hence contains machine-dependencies. However it has retained the basic attributes of SAIL as an extended ALGOL-like language. Among MAINSAIL's language features are:

machine-independent language design
straightforward syntax and semantics
efficient code generation for variety of machines
separately compiled segments
double precision integer and real
bit and address manipulation
variable-length strings
dynamic and static records
generic procedures
default and repeatable arguments
static initialization
in-line assembly language
macro facility
compile-time evaluation
conditional compilation
multiple source files during compilation
sequential and random i/o
terminal interaction
comprehensive system procedures
mathematical library
access to dynamic storage allocation
access to runtime system
"garbage collection" of strings and records.

MAINSAIL is designed as a programming system rather than just a programming language. It is presently composed of a compiler generator, a compiler, and a runtime system. Further components envisioned are a debugger, a code optimizer, and a text editor. All of these components are to be written in MAINSAIL, and hence made fully portable. Also, there are plans for extensions to the MAINSAIL language, such as:

coroutines
extensible data types
extensible code generation
list processing system (LEAP).

In its role as a research tool, MAINSAIL is being used for an examination of the following design issues:

language design --- what syntactic and semantic constraints are imposed by portability.

compiler design --- how can a single language be compiled into efficient code for a large number of computers.

runtime design --- to what extent can the runtime system be made transportable.

computer design --- what architectures best support a high-level language implementation.

program design --- what role can portability play in the design of reliable software.

Since the PDP-10 and PDP-11 implementations will be the first to become available, this user community has been introduced to MAINSAIL at several meetings of DECUS, the Digital Equipment Corporation Users Society. MAINSAIL was first described in a talk delivered at the DECUS DECsystem10 Fall '75 symposium in Los Angeles. It was then featured in a session entitled "Languages for Portability", at the DECUS DECsystem10 Spring '76 symposium in Hyannis Port. A paper will be presented at the DECUS mini/midi Spring '76 symposium in Atlanta. These sessions are resulting in an almost continual stream of inquiries concerning MAINSAIL.

Before MAINSAIL is exported to other sites, it will be thoroughly tested on several local computer systems. It is now being used on a PDP-10 with TENEX and a PDP-11 with RT11. Implementations for a PDP-10 with TOPS-10 and an IBM-370 with ORVYL (Stanford operating system) are now under development. Code has also been generated for an INTERDATA-716 and a Data General NOVA, both mini-computers. There is interest in using MAINSAIL on a PDP-11 with operating systems such as UNIX and RSX-11; a DECsystem20; and the NIH-GPP, a parallel processor being constructed for the NIH Image Processing Unit. MAINSAIL will be made available on such machines as sufficient funding is obtained to support an expanded effort.

A number of projects are interested in using MAINSAIL for the development of portable software. Among these are:

robotics project
mass spectrometry system
computer-aided-instruction system for the teaching of logic
automated cell classification laboratory
machine-independent version of (a subset of) INTERLISP
display-oriented text editor (TV-EDIT)

Several INTERLISP programs are being considered for translation into MAINSAIL. These programs have developed to the point that they no longer require the very general environment supported by INTERLISP, and hence can avoid the related inefficiencies. Also, there is interest in providing such programs to a wider community with systems other than a PDP-10 with TENEX.

MAINSAIL can furnish the capabilities necessary to develop generally useful software, and to distribute that software to a wider community than any existing language can now reach. To reach that goal, several issues must be resolved.

First, it must be demonstrated that MAINSAIL can be efficiently implemented on a wide variety of computer systems. The creation of new implementations must be largely automated, with the machine-dependent aspects requiring at most a few man-months for development. Some plausible target systems are:

```
PDP-10          (TENEX, TOPS-10)
PDP-11          (RT-11, RSX-11, RSTS, UNIX)
DECsystem20
IBM-360/370     (ORVYL, TSS, TSO)
NOVA/ECLIPSE
additional large machines: CDC, UNIVAC, ...
additional small machines: INTERDATA, TI990, MICRODATA, HP-3000, ...
```

Second, a suitable means of distributing the MAINSAIL programming system, along with portable software written in MAINSAIL, must be determined which is consistent with resources available to SUMEX. Steps must be taken to insure compatibility among the various sites, and to minimize maintenance of the machine-independent parts. The development and distribution of MAINSAIL is viewed as a time-consuming process fraught with complications. SUMEX must be careful not to make promises which cannot be kept.

Finally, program developers must be motivated to design machine-independent systems, to promote their distribution, and to use portable software developed at other sites. Machine-dependencies must be eliminated, or at least isolated and documented. Program design must face, from the onset, the restrictions imposed by portability.

II.A.2.g OPERATIONS AND USER SOFTWARE

Operations Programs:

The programs which assist system operations and management have been effectively organized this past year. A catalog has been made of approximately 40 operator programs which had been maintained by various staff members to facilitate their particular tasks and passed along to other staff members informally only. Some of the purposes of the new operator utilities include: setting the GUEST password and the access of guests to other user programs (only restricted access is allowed for guests); providing summary information on directories and groups;

exercising various hardware devices; performing error analysis for system crashes; changing system downtime information; creating new directories and setting the appropriate parameters for each user; sending important notifications to logged in terminals; watching terminal activity; and handling special resource allocation situations such as project demonstrations. Many operator tasks have also been automated as "autojobs" or batch jobs which run on the system performing tasks continuously or on predefined time schedules. In addition to this collection of basic operator utility programs, software has also been improved for system-wide statistics gathering, accounting (including information on diurnal community loading and service), disk allocation checking and enforcement, system backup onto tape for file protection (any file in existence for 24 hours will now be guaranteed retrievable for a two-month period), and currently a new spooler is being written for the lineprinter to provide faster handling with more control of the printer queue available. Another development effort is in the area of collecting and organizing statistics about subsystem use. Such data can help in planning where to concentrate development effort, determining which users to notify about new program information, and getting users and user groups with similar interests together.

User Software:

As the user community has stabilized certain trends in user needs have been noted and utility software has been collected and written in those areas. Particular attention was also paid this year to making the user interfaces for all existing utilities as consistent as possible and many programs received minor cosmetic modifications.

STATISTICS PACKAGES

Three new utilities packages were added: STP (STATPACK Statistical Package) and BANK (a data management package) both from Western Michigan University; and SPSS (Statistical Package for the Social Sciences) converted by University of Pittsburgh for PDP10 use (and obtained from them under contract).

DEC SOFTWARE

There is a new DEC policy to issue maintenance releases at least every two years for all software. This has resulted in a large volume of new DEC releases over the past year. Essentially we receive DEC software under three separate arrangements. First, FORTRAN10 and LINK10 were purchased under contract. Second, we have subscribed on an annual basis to the DEC distribution tapes from which this year, we put up new versions of RUNOFF, PIP, BASIC, BLIS10, CCL, FILEX, CREF, CHANGE (known at SUMEX as DCHANGE due to name conflict), FLECS, GLOB, MTCOPY, MACRO, TECO, and LOADER. A variety of bug fixes were done on these programs and sent back to DEC. Also, the SCAN program was modified for handling TENEX-style directory names for these programs. For most of the programs, after some practice and with the help of a TECO routine to automatically check the stored SRCCOM record of local modifications to the old versions, it became

a one or two day effort to get each program up. The third source of DEC software is from the DECUS tapes which are ordered individually with a minimal per tape charge. We order all programs from these tapes which appeared to be possibly useful. The tapes are then stored until a user request is actually received for a given program. The only such request to date has been for SIMULA.

PA1050 EMULATOR

All of the above official DEC software plus an assortment of other programs imported from TOPS10 sites can only be run through the use of a TOPS10 emulator - a program which converts all TOPS-10 system calls (UUO's) into "equivalent" JSYS TENEX calls. The PA1050 emulator is a standard part of TENEX software. However, it is badly out-of-date and inadequate for many complicated programming situations. SUMEX has developed its own version of PA1050 (incorporating features of other local versions wherever possible). The SUMEX PA1050 has now received reasonably wide distribution to other TENEX sites. Modifications to PA1050 this year included: conditional code for SRI and IMSSS use of PA1050, a change in the standard altmode character, additions to the terminal input/output routines, a provision for running FORTRAN jobs detached and a number of other changes to accommodate the new FORTRAN10. Also added were capabilities to assign device names to common directories to facilitate easy access, additions required by the plotter, debugging of buffer synchronization, improvements in the pseudo-interrupt (PSI) handling, and other assorted bug fixes.

EDITORS

A variety of editors are offered so that users will have a choice and to make available any editor that a user may be accustomed to from another PDP10 system. In general, only SOS or TECO for hardcopy terminals and TV for DataMedia display terminals are widely used. We have the TENEX version of TECO and also added the DEC TECO this year. The transition from our local SOS to UTAH-10 SOS is proceeding smoothly and should be completed this month. Error reports and suggestions for new features were solicited from all SOS users and passed along to KKAY@UTAH-10. Similarly, we have made a recent effort to determine the SUMEX community priorities for development of TV; and we are currently conferring with Mr. Pentti Kanerva of IMSSS regarding the future direction of TV development.

BATCH PROCESSING

With the increasing system load, it is advantageous to both the user and the system to perform as much work in non-peak hours as possible. The BATCH facility allows submission of non-interactive jobs to be spooled for later automatic running. BATCH required extensive debugging and modification this year but is now fully operational. Users are being strongly encouraged to consider this option.

MESSAGE PROGRAMS

Bug fixing and streamlining of the SNDMSG (mail sending facility) has continued. We have also adopted a new standard mail reading facility called MSG which is authored/maintained by Mr. John Vittal of the Information Sciences Institute (ISIB). Extensive communication with Mr. Vittal in the developmental stages has lead to a product which serves the SUMEX community needs very well and runs with no problems after local incompatibilities were diagnosed and accommodated in the MSG program.

SEARCH PROGRAMS

A new fast substring search program, XSEARCH, has been written in SAIL by Mr. Scott Daniels of IMSSS/SUMEX. This has been highly successful as a stand-alone search program. It can look through the entire <DOC> directory in about a minute of CPU time. The core code of XSEARCH has also been worked into two library packages: USEARCH which stresses flexible printing of the context of the found string (and is used as the basis for the new HELP program) and PSEARCH which uses TENEX PMAP'ing to increase the search speed even more. A similar very fast algorithm has been incorporated in the WHOIS program by Dr. Lederberg. WHOIS is an often-used program for searching a file containing user names, addresses, and affiliations and the speed-up from the better search algorithm has been a major improvement.

OMNIGRAPH

SUMEX has added a CALCOMP plotter, a Tektronix terminal, and a GT40 light-pen facility to the locally available hardware. The software for the Tektronix was already available in the OMNIGRAPH package from NIH. SUMEX did write a local demonstration program for the cross-hair feature of the Tektronix. The CALCOMP plotter was also available with OMNIGRAPH; but due to differences both in the facilities offered with the particular terminal and in the TOPS10 and TENEX environments, a large scale three-fold programming effort was necessary to get our CALCOMP running. First, a spooler is being written for the plotter (independent of OMNIGRAPH). Second, the PLOTX program of OMNIGRAPH has been debugged and extended for local use: online access is allowed, plot titles are used, x and y may be stretched independently, a clipping routine is added, and the SAIL record data structure is used so that the limit on number of plots is removed. Third, a general CALCOMP control load module was written which currently is used to drive PLOTX but could also be used as a general plotting facility to form the basis of other plotting packages. This module adds more extensions to the plotting capabilities: string prints and line plots with edge checking and arbitrary specification of character codes.

A light pen facility has been added to the OMNIGRAPH code for the GT40 by Mr. Frank Wingate or our staff. This work was done in conjunction with NIH and the original design was followed with all code being incorporated back into the NIH master source files for OMNIGRAPH. A demonstration program was also written for this new feature.

DISPLAY TERMINAL SUPPORT

An increasing number of users have access to Datamedia display terminals. In addition to the TV editor, a variety of programs have been written specifically for these terminals. EDIR places a representation of the user's directory on the screen and allows him/her to point the cursor at a file and give a command letter to indicate desired actions such as Archive, Delete, Undelete, Type (which clears and later refreshes the screen), List, etc. The screen is updated to reflect the current state of each file and a running total of active and deleted file pages is displayed at the top of the screen.

SYSMON displays system loading statistics on the CRT screen and updates the display at regular intervals to reflect changes. included in the display is a ranking of active users according to CPU time used and statistics on i/o use, size of the balance set, idle time, load average, etc.

SCROLL is an editor for creating and storing pictures which can then be reprinted on the screen at any time under program control. The editor facility is especially designed for moving freely among the parts of a picture and is particularly suited for flow-chart drawing. The ability to call pictures which are stored under a given name is very useful in CAI work and other demonstrations where predefined displays can be used as illustrations.

RECORD

The RECORD program written in SAIL by Dr. Robert Smith has become very popular at SUMEX this year and also has been modified to meet the needs of the DENDRAL group for dealing with their GUEST users. RECORD was designed to use the pseudo-teletype facilities for making a file typescript of a terminal session. It has been extended so that the PTY job, once started, can run independently freeing the physical terminal for other work. In the new DENDRAL version, a guest user logging-in on a special directory is automatically interfaced to RECORD to make a typescript of the session (with the user's knowledge). This entire operation is transparent to the user and facilitates later analysis of the session by the DENDRAL staff to learn where the programs need improvement.

PUB

PUB (written in SAIL by Mr. Larry Tesler formerly of SU-AI and currently at XEROX-PARC) is a very powerful documentation preparation language. It is more difficult to use than DEC's RUNOFF but also is more flexible. Last year SUMEX produced a substantially improved set of macros that make use of PUB simpler. This was accompanied by a new manual and a series of well-received PUB introductory classes. New extensions to the PUB macro facility this year include: automatic bibliographic entries from a library with flexible cross-reference printing formats, automatic queuing and placement of tables and figures, multi-column handling, and a marginal notation feature allowing revised or otherwise emphasized text to

be marked. Requests for the SUMEX macro package for PUB have been received from NIH, SRI, AMES, ISI, and IMSSS.

DIABLO

This program is a driver for terminals with a DIABLO or DIABLO-like print mechanism which are used to produce high quality typed output. The DIABLO program is specifically designed for printing PUB output files and it will handle PUB underlining and half-line printing. DIABLO now supports GEN COM as well as DTC terminals. The PUB/DIABLO combination has been upgraded to print a form of proportional spacing which consists of evening the spaces between words when justifying to the right margin. This produced a significant improvement in the appearance of the output. Plans are also on-going to include a hyphenation algorithm in PUB which will be another significant improvement.

TAPE PROGRAMS

Foreign tapes continue to cause time-consuming problems in transferring data to our machine. New tape documentation has been written which explains the tape situation to users and makes recommendations about which of the various tape reading programs to use for various format tapes. Two new tape programs have also been added. MTCOPY from DEC has the ability to handle multiple tapes in a single command. The SUPXEC program by Mr. Ron Roberts of IMSSS was designed to operate like an EXEC for tapes with complete Directory, Copy, and Delete commands available.

A large variety of smaller utilities programs have also been added to the SUMEX catalog. With the file system operating near capacity, one emphasis this year has been on file management programs. Utilities are now available: to plot the age of a user's files, to allow deletion/archiving of all files older than a cut-off date, to find all files newer than n hours, to clean up wasted file space in text files, to find files with multiple versions, to rename files, to copy selected portions of files together, to view the actual character codes in a file, to check the file-descriptor-block (FDB) for a file, to find all new public files, to find the exact location of a file, to recognize partial filenames, to print files in reverse order, to provide a handle on files too large to edit conveniently, to encrypt text files, to convert the character case of files, etc.

Another area of utilities development has been programs to manage personal calendars or to serve as on-line reminder systems. And, of course, an area of continuing development is informational utilities. The WHOIS program to give name/address/phone number and project affiliation information on users may well be the most-often used of these utility programs.

II.A.2.h USER ASSISTANCE AND CONSULTING

User consulting continues to play a key role at SUMEX. Because of the geographic distribution of our users where they may have little or no direct contact with computer experts and the nature of the user community in which many non-computer professionals are involved, the user consulting load is higher than on most similar systems. While direct individual consultation has been a major component of the effort and will continue to be, other solutions to the problem are continually sought. A number of approaches have proven useful:

1) Foster interactions among users themselves to help each other learn and to solve particular problems. This has been the case with the new statistical packages. The staff is available to fix program bugs but in fact has little expertise in the use of the packages. So the groups expressing interest have been put in touch with each other. This effort has been quite successful.

2) Among the systems coupled by networks (ARPANET currently), focus maintenance responsibility for particular pieces of software in the groups which developed them or where an extensive expertise already exists. Typically the author/maintainer is best able to deal with user problems. With the ARPANET to provide communication access at a non-local site, the duplicated effort in a local staff trying to familiarize themselves with imported programs for bug fixing and consulting is minimized. An example of this is the shift this year from our local SOS editor program which had been developed by a former staff member and was no longer actively supported by any current staff member to a version of SOS which is well-maintained at UTAH-10 by Kevin Kay. He analyzed our version, incorporated the improved features into his version, and assumed support of it at SUMEX. This is an increasingly common phenomenon among the TENEX sites on the ARPANET. Other software maintained by one site for all the sites includes MSG, REDUCE, SAIL, INTERLISP and PUB. In general, one local staff member is the primary contact for each of these and does handle routine problems and questions but does so in close communication with the author. And in fact, for both SOS and SAIL, users are encouraged to go directly to the authors; SOS has a built-in GRIPE facility which sends the user message to SOS@SRI.

3) Employ media which reach a large number of users rather than dealing on an individual basis. This includes writing of documentation (see Section II.A.2.j on page 40) and holding classes and tutorials. Last year, several classes were given (PUB, SAIL, and machine language). These were successful; and we have followed up with an advanced SAIL course, a BAIL lecture, and a planned repeat of the PUB class. More informal meetings have also been held to discuss issues such as choice of a programming language and efficient programming in languages such as SAIL and INTERLISP. Participants in last year's Workshop requested an INTERLISP tutorial which Dr. R. Carhart of the DENDRAL project gave. A number of tutorials on languages and other aspects of SUMEX use are planned for the June 1976 Workshop.

4) Use other techniques for interactive on-line teaching. Interactive

computers have been used in a number of areas for computer-assisted-instruction (CAI) applications. We may be able to adapt some of these techniques and will be studying possibilities of a training mode for selected programs (possibly a text editor or the EXEC). In this we are fortunate to have close ties with IMSSS, a well-known leader in the field of CAI.

II.A.2.i INTRA-COMMUNITY COMMUNICATION

Help System:

A substantial problem for users not intimately familiar with the system (and even those who are) is how to locate the appropriate documentation on-line to answer a particular question as it arises. Many times a staff member or other user is not available to help so we have been developing various forms of on-line assistance or "help". After considering a number of help schemes, early this year we put up a temporary help program which optionally printed a general information file and then interactively helped the user search through the names of the <DOC> files with a keyword search. This approach is rather effective at SUMEX where long filenames composed of as many keywords as possible are used to identify the information files. The interim help program was well-liked by users. A new more sophisticated version is currently being tested.

It offers all the facilities of the simple version but it also has extensions in three basic directions. First, in addition to checking the filenames for the given keyword, it also checks the contents of several general information files, e.g., the file listing all programs with a one-line summary of each, a file containing an on-line index of the jsys's, a file containing entries on new programs, and a file containing the TENEX command equivalents for TOPS10 commands. If the keyword is found in the keyword line of any of these entries then the individual entry is printed out. These files were all existing documentation files which needed only slight format changes to be used in this way. Second, the user can specify certain standard keywords which allows him to do more specialized searching of the relevant data base for the topic. Third, when the filenames of the <DOC> directory are searched for the keyword and a list of possibly relevant files has been produced then the user has several options. The contents of any of the files can be searched for further keywords or the entire files (or selected pages) can be typed on the terminal, put together in a new file, or listed on the printer. This allows the user to browse around and tailor-make a new document with just the desired pieces found.

Bulletin Board System:

Another kind of user communication system has been under design and implementation for some time at SUMEX. Some information is of a more informal or transient nature than that comprising files suitable for the <DOC> directory. Other types of information have relevance for only a subset of the community such as intra-project communication about program design, external users, etc. We have maintained a <BULLETINS> directory for system-related "bulletins" (see page 228 for a current directory listing) but have needed a more general facility serving public and private bulletin boards as well as allowing users to selectively direct their interests to particular subsets of the information. Such a bulletin-board system would fill the gap between the <DOC> directory, intended for permanent documentation, and the mail system, where each user (and the system) has his own mailbox. It would complement the help system, providing quick access to such intermediate information.

An on-line "bulletin board" system is nearly ready for release to the SUMEX community. The following is a brief description of the system in its preliminary implementation. We expect it to grow and change as users begin to use it and identify additional communication needs.

The system has a number of bulletin boards available. A public bulletin board encompasses system-wide information. It is for community use and has bulletins on new features of system programs, announcements, corrections, progress reports, suggestions, queries, and comments of general interest to the SUMEX community. There may be need for other public bulletin boards such as for the AIM workshop depending on the overall volume of information involved. Private bulletin boards may focus on individual projects (e.g., DENDRAL, ONET, MOLGEN, etc.), subsystems, or other subgroupings of the community.

Bulletins are filed on the bulletin boards under topics, which may have any number of subtopics. They each have an expiration date, because some information may be of a temporary nature. In general, the kind of bulletin posted is what used to be sent out in multiple copies with a SNDMSG distribution list to a number of users, and would now be posted on the bulletin board pertaining to that interest group or project.

Two programs operate on the bulletin-board files. POST is the equivalent of SNDMSG or ADDMSG with extra editing and display features and doubles as a bulletin poster. POST can send copies of the same bulletin to a user list and bulletin boards at the same time. When not used to post a bulletin, it behaves like SNDMSG. The program BBD performs other inquiry and editing functions on bulletin boards. It is designed to behave as the TENEX EXEC does, for consistency with existing command conventions and ease of use. There are also some similarities between BBD and message-reading programs like MSG and BANANARD. A BBD user will be able to connect to the bulletin board of his choice and:

Get a directory listing of topics and bulletins

Type a (set of) bulletin(s) by number or topic name

Ask to see the first 5 lines only of each

Copy bulletins to other files (including TTY: or LPT:)
in message format

In each of the above, the user can narrow his range of bulletins by asking BBD only for those that meet certain criteria, such as:

New bulletins

Bulletins filed under topics on his "interest list"

Bulletins written by a particular author or group of authors

Deleted bulletins

Expired bulletins

Bulletins posted before or after a specified date

Bulletins with a desired phrase in the message or subject line

or combinations of the above.

There will be a notification system whereby bulletin-board users are notified once per day of new bulletin arrivals. The user tells BBD to add a given topic to his "interest list". He is notified of new arrivals only for those topics, unless his interest list is "all topics". There is a separate interest list for each bulletin board. The notification system can easily be extended to be an automatic reminder system. One could post a bulletin on a "Reminders" bulletin-board. The bulletin would be sent to those to whom it was addressed when it expired.

Bulletin-board users will likely want BBD to assume a few things for them. They may specify a bulletin-board to connect to upon entering the program (default is the main bulletin-board), or they may never want to see anything but what is on their interest list. BBD will have a "Defaults" command which will set these things up on a per user basis.

Each bulletin board will have a manager, who has special privileges. He will be able to create and destroy topics, delete and undelete bulletins, reorder the bulletins in each topic, change the assignment of bulletins to topics, and change expire dates. Everybody will be able to delete, undelete, refile, and change the expire dates of bulletins they author.

A first version of the bulletin board system is in final checkout. We expect to release it to the community this summer and to continue development based on user reactions and needs during the upcoming year.

II.A.2.j DOCUMENTATION AND EDUCATION

SUMEX has set and maintained high standards for documentation. This year we achieved virtually complete documentation of all available programs. The list of the <DOC> directory reported last year contained 142 files. That has now increased to 220 files (see Appendix F on page 218); many of which have been updated and reorganized. All of the general information files have also been updated during the course of the year. SUMEX is probably the best documented PDP10 site on the ARPANET. However, we have long recognized the fact that the usefulness of the documentation is severely limited by the ease of access to the particular bit of information that a user is currently seeking. As the volume of documentation increases, more information is available in an abstract sense but may perversely become less available in real terms because of the difficulty of finding it. The HELP and BULLETIN BOARD systems are designed to help overcome these problems.

This year, SUMEX submitted a 25-page entry to the ARPANET Resource Handbook (September 1975) which contains a variety of information on the ARPANET host systems. Our entry includes a description of the SUMEX facility and projects, a list of the software available for export with a policy statement on the procedure for obtaining this software, and a summary of the major areas of interest at SUMEX. A copy of this material is attached as Appendix G on page 230.

As a follow-up to last year's very successful SAIL introductory classes, this year a series of more advanced SAIL and Machine Language seminars was given by Dr. R. Smith. These covered the interface of SAIL to the PDP10 host machine and timesharing system, SAIL program debugging, and implementation and efficiency considerations. Mr. John Reiser of SU-AI gave a seminar on the features and use of the new SAIL debugging system, BAIL. A SAIL Tutorial has been written by Dr. Nancy Smith of SUMEX which will be published shortly by SU-AI along with a reprinting of the standard SAIL document plus the TENEX-SAIL Manual and the new BAIL Manual. Finally, Dr. Nancy Smith, in conjunction with the improved macro facilities and documentation, gave an introductory class in using PUB.

II.A.2.k SOFTWARE COMPATIBILITY AND SHARING

Over the past year, in our commitment to software importation where possible rather than reinvention, we have encountered numerous experiences in the sharing of software. At SUMEX many avenues exist for sharing between the system staff, various user projects, other facilities, and vendors. In the past without communication networks, the system vendor served as the focal point for distribution of most software to user sites. Since the process of distributing tapes (and particularly of handling bug reports and user suggestions) was very slow, it was common for sites to take a version of a program and then modify and maintain it locally. This caused a proliferation of home-grown versions of software. Similar impediments have existed to the dissemination of user software. User organizations like SHARE and DECUS have helped to overcome these problems but communication is still cumbersome. The advent of fast and convenient communication facilities coupling communities of computer facilities has

the potential of making a major difference in facilitating inter-group cooperation and to lower these barriers.

Recently, the TENEX sites on the ARPANET have been interacting increasingly with each other to develop new software systems. This functions effectively to build communication around the network and promote a functional division of labor and expertise. The other major advantage is that as a by-product of the constant communication about particular software, personal connections between staff members of the various sites develop. These connections serve to pass general information about software tools and to encourage the exchange of ideas among the sites. Certain common problems are now regularly discussed on a multi-site level. We continue to draw significant amounts of system software from other ARPANET sites reciprocating with our own local developments.

Currently the number of sites involved is relatively small and the interactions are informal. It may be that this informality is an essential ingredient to making this process work, much as friendships among people develop. It may be the bureaucracy of vendor systems and procedures (which do have useful fallout in uniform documentation, interfaces, etc.) which caused the proliferation of home-grown systems in the past. Indeed our own attempt at building a SAIL library may have foundered because we tried to be too formal about it.

We began an effort last year to accumulate useful SAIL library routines from the various groups which have been working with this language (Stanford AI, IMSSS, SRI, NIH, USC-ISI, etc.). It has been somewhat surprising that so little communication of SAIL library programs has taken place - it is almost literally true that each user has his own stock of tools in private procedure libraries. We sent a letter to interested groups soliciting inputs on a basis which attempted to balance the problem of assuring library quality and integrity against establishing so high a threshold for quality and polish that individuals are not motivated to cooperate. Despite the willingness of active community to share time and ideas on an individual basis, there have been virtually no external entries to the library in response to our efforts.

In other areas, however, where individuals have undertaken the design of major software components, mutual design cooperation between sites has a growing list of examples of success. Undoubtedly the particular personalities involved play some role as well as the orientation of the funding agencies.

Certainly the TENEX operating system itself is an example of community cooperation although there has been some tendency for localization because of BB&N's rigidity. Other examples of cooperation mentioned earlier include SOS, MSG, PUB, the batch system, and our substantial efforts to contribute to software exportability through developing the MAINSAIL system. This latter effort has received very enthusiastic support from many quarters of the computer community. Other noteworthy examples encountered this past year include the following.

When Mr. John Reiser began writing the BAIL debugger for SAIL, he

was contacted and agreed to design the program for maximum compatibility between the SU-AI system, TENEX, and standard TOPS10. This effort was quite successful. BAIL was written in such a way as to use each of the operating systems optimally with no compromise in program design. One estimate of the extra development time involved was only 10 per cent. The important ingredients are complete program comments, modular design, and no unnecessary system dependent code. It can be contrasted with other programs written at SU-AI. For example, in the process of designing a bulletins system, SUMEX learned of the AP NEWS Service program at SU-AI which has many features similar to our design for bulletins. The program was studied and proved to be very difficult to transfer and adapt because of the choice of language and the degree of dependence on the SU-AI home-brew operating system. Both BAIL and NEWS were written at SU-AI and both are well-written programs by other criteria.

MLAB and OMNIGRAPH -- Even without the facilities of the ARPANET and with all the compatibility problems of TOPS10/TENEX program sharing, our interactions with the NIH Division of Computing Research and Technology concerning MLAB and OMNIGRAPH have been mutually beneficial. SUMEX has sent code for a "TENEX" conditional compilation switch to NIH which has been incorporated into the source files. Also, the light pen and plotting development work done here this past year has had close communication with NIH.

With very careful organization by Dr. R. Smith, the export of TENEX-SAIL has proceeded with very good community cooperation. A special directory called <XSAIL> has been created (which can be accessed by the ANONYMOUS feature of the FTP file transfer program so that there is no need for exchanging passwords). This directory contains ALL the necessary files for exporting the SAIL system. All sites running TENEX-SAIL were individually contacted and requested to appoint a local contact for communications purposes.

The openness of the ARPANET communication facilities encourages some members to copy pieces of software without the author's knowledge thereby defeating the necessary more orderly processes of maintenance and upgrade and in some cases losing proper attribution for the program's development. This has occurred with a number of the programs that we have available for export. It is difficult to control such behavior without at the same time limiting access for cooperating members of the community. We try to discourage it by pointing out the self-defeating effects.

Another continuing effort is in the maintenance of software compatibility with DEC. The PA1050 program (for TOPS10 emulation) is an important part of the software for each TENEX site. SUMEX made a search for all local versions of PA1050 and combined the best features. Much new work was also done. This version has been made available to other TENEX sites - IMSSS and SRI are running it with direct cooperation and other sites have copied it without informing us. Since almost all sites using TENEX are doing government-funded work and this is an obligatory condition for ARPANET access, we have not felt it necessary until now to take strenuous (and possible costly) measures to protect this software. We will, however, review this problem periodically.

A new aspect of DEC compatibility arises with the announcement of the TOPS-20 operating system which has been developed by DEC for their KL-20 machine. The current 2040 system is a relatively small system but will be followed by larger members of the 20-family. TOPS-20 is based on TENEX. It appears that ARPA may be transferring support from BBN to DEC for system development and the ARPANET interface for the system. This has the potential for greatly decreasing the compatibility problems since both TOPS-10 and TOPS-20 will be under DEC control. On the other hand, DEC has implemented a variety of minor changes already (new JSYS's, different file name notation, etc.) which are causing a divergence between TOPS-20 and TENEX that may well lead to greater compatibility problems than exist now. We noted these possibilities in the decision to remain with TENEX and implement the dual processor system. The timing of DEC's evolution of TOPS-20 with larger scale processors is uncertain as is the rate with which the ARPANET community might move in that direction. There are many existing KA-10 and KI-10 machines running TENEX for which there are no current prospects of replacement. Over the next few years we feel our decision was correct, especially in view of budgetary constraints. However, we are sensitive to remaining as parallel as possible with the mainstream of the community and will actively pursue this goal.

II.A.3 RESOURCE MANAGEMENT

Over the past year, the SUMEX project has devoted a substantial part of its effort toward its community-building role in recruiting new projects, promoting interactions between user projects, and encouraging dissemination of running performance programs to medical scientists. The following summarizes specific aspects of SUMEX-AIM community management activities.

II.A.3.a MANAGEMENT COMMITTEES

The SUMEX-AIM resource is constituted to attempt to bring into closer contact collaborating health research groups from around the country. This mission entails both the recruitment of appropriate research projects interested in medical AI applications and the catalysis of interactions among these groups and the broader medical community. As this effort is not a unilateral undertaking by its very nature, we have created several management committees to assist in administering the various portions of the SUMEX resource. As defined in the SUMEX-AIM management plan adopted at the time the resource grant was awarded, the available facility capacity is allocated 40% to Stanford Medical School projects, 40% to national projects, and 20% to system development and related functions.

Within the Stanford aliquot, Dr. Lederberg has established an advisory committee to assist him in selecting and allocating resources among projects appropriate to the SUMEX mission. The current membership of this committee is listed in Appendix H.

For the national community, two committees serve complementary functions. An Executive Committee oversees the operations of the resource as related to national users and makes the final decisions on authorizing admission for projects. It also establishes policies for resource allocation and approves plans for resource development and augmentation within the national portion of SUMEX. The Executive Committee oversees the planning and implementation of the AIM Workshop series and assures coordination with other AIM activities as well. The workshops are being carried out under Dr. S. Amarel of the Rutgers Computers in Biomedicine resource. The current membership of the Executive committee is listed in Appendix H.

Under the Executive Committee functions an Advisory Group representing contact with medical and computer science research relevant to AIM goals. The Advisory Group serves several functions in advising the Executive Committee; 1) recruiting appropriate medical/computer science projects, 2) reviewing and recommending priorities for allocation of resource capacity to specific projects based on scientific quality and medical relevance, and 3) recommending policies and development goals for the resource. The current Advisory Group membership is given in Appendix H.

These committees are actively functioning in support of the resource. Meetings to date have been held by telephone conference for the most part owing to the size of the groups and to save the time and expense of personal travel to meet face to face. These "missings" (a term coined by Dr. Licklider), in conjunction with terminal access to related text materials, have served quite well in accomplishing the agenda business and facilitate greatly the arrangement of meetings. A few technical problems occasionally attend such sessions such as poor telephone reception for some members but in general this approach is quite satisfactory. The key to success seems to be a) fairly short and not too infrequent sessions, b) a firm agenda, c) mail distribution of relevant documents, d) computer network backup for exchange of information, and e) informality and personal rapport of the members. Other solicitations of advice requiring review of sizable written proposals are done by the mails.

II.A.3.b NEW PROJECT RECRUITING

As a result of the public announcements of the SUMEX resource, NIH contacts with prospective grantees, and personal contacts by the staff or committee members, a number of additional projects have been admitted to SUMEX; others are working tentatively as pilot projects or are under review. We have prepared a variety of materials for the new user ranging from general information such as is contained in the brochure (Appendix I) to more detailed information and guidelines for determining whether a user project is appropriate for the SUMEX-AIM resource. Dr. E. Levinthal has prepared a questionnaire to assist users seriously considering applying for access to SUMEX-AIM (see Appendix J). Pilot project categories have been established both within the Stanford and national aliquots of the facility capacity to assist and encourage projects just formulating possible AIM proposals pending a formal review.

The projects newly admitted over the past year include (see Section IV for more detailed descriptions):

National -

1) Chemical Synthesis Project (SECS); Dr. T. Wipke (University of California at Santa Cruz)

2) Language Acquisition Modelling (ACT); Dr. J. Anderson (University of Michigan)

As an additional aid to new projects or collaborators with existing projects, we have a limited amount of funds which are being used to support terminals and communications needs of users without access to such equipment. We are currently leasing 6 terminals and 4 modems for users as well as 4 foreign exchange lines to better couple the Rutgers project into the TYMNET and a leased line between Stanford and U. C. Santa Cruz for the Chemical Synthesis project.

II.A.3.c STANFORD COMMUNITY BUILDING

During the past year, the Stanford community has undertaken several efforts to encourage interactions and sharing between the projects centered here. Beginning in the fall term, Professor Feigenbaum organized a seminar class with the goal of assembling a handbook of AI concepts, techniques, and current state-of-the-art. This project has had enthusiastic support from the students and substantial progress made in preparing many sections of the handbook. An outline of the material to be prepared along with an indication of the status of each article can be found in Appendix B on page 180. Several examples of completed articles are given in Appendix A on page 166.

A second community-building effort was a "mini AI conference" held at Stanford in January 1976. This 3 day series of meetings featured presentations by each of the local projects and comparative discussions of approaches to current problems in AI research such as knowledge representations, production system strategies and rule formation, etc. A brief summary of the conference is attached as Appendix C on page 194.

II.A.3.d RESOURCE ALLOCATION POLICIES

As the SUMEX facility has become increasingly loaded, a number of diverse and conflicting demands have arisen which require controlled allocation of critical facility resources (file space and central processor time). We have already spelled out a policy for file space management; an allocation of file storage is defined for each authorized project in conjunction with the management committees. This allocation is divided among project members in any way desired by the individual principal investigators. System allocation enforcement is implemented by project each week. As the weekly file dump is done, if the aggregate space in use by a project is over its allocation, files are archived from user directories over allocation until the project is within its allocation.

As described under TENEX monitor software development, we have been using a primitive CPU scheduling algorithm intended to ensure that no one user gets more than a fair share of the machine when other users are contending. With the implementation of TENEX 1.33 this summer, the pie-slice allocation system will be available to more rigorously ensure CPU allocation by project and community allocations.

As also mentioned earlier, we have categorized users in terms of access privileges. These comprise fully authorized users, pilot projects, guests, and network visitors in descending order of system capabilities. We want to encourage bona fide medical and health research people to experiment with the various programs available with a minimum of red tape while not allowing unauthenticated users to bypass the advisory group screening procedures by coming on as guests. So far we believe we have

had little or no exploitation compared to what other sites have experienced, perhaps on account of the personal attention that senior staff gives to the logon records. However, the experience of most other computer managers behooves us to be cautious about being as wide-open as might be preferred for informal service to pilot efforts and demonstrations. We will continue developing this mechanism in conjunction with management committee policy decisions.

II.A.3.e AIM WORKSHOP SUPPORT

The Rutgers Computers in Biomedicine resource (under Dr. Saul Amarel) is actively working on plans for the second AIM workshop this June. The current plans call for a four day series of meetings covering a range of topics related to artificial intelligence research, medical needs, and resource sharing policies within NIH. The SUMEX facility will act as a prime computing base for the workshop demonstrations. We hope to have the new dual processor system in operation for the meetings. A final decision will depend on progress over the next week in completing the debugging of the initial system and our ability to assure reliable operation. We are in the process of working with Rutgers to provide backup modes for program demonstrations in the event of computer system problems.

II.A.4 FUTURE PLANS

System Development:

In the next year much work remains to complete the dual processor system. We must complete installation, evaluate its performance in terms of increased throughput, identify and fix excessive waiting for monitor interlocks, and optimize system scheduling and resource handling. We plan to implement a mutual interrupt facility between the two machines and to implement a bus switch allowing I/O devices to be moved easily between the two machines. This will increase our ability to keep the system running in the case of a processor failure by reconfiguring to a single processor mode.

We plan to continue evaluation of system hardware bottlenecks and to pursue avenues to eliminate them. We know that disk space is currently a problem and are trying to augment the system through user project cooperation. Other limiting resources over the next year may be memory and swapping space.

We will install version 1.33/1.34 of TENEX with necessary dual processor and KI-10 modifications in order to stay current with other TENEX sites and to improve resource allocation controls among the AIM community members.

We plan to improve the batch processing capability for those jobs which need not run interactively. A current system has helped to move system loading from prime time to off hours. We plan to extend facilities for error handling and more flexible job scheduling.

We will continue to refine the Executive program and capabilities for guest users in conjunction with the TENEX 1.33 upgrade.

We will also investigate ways of improving network communication services. This will include attempts to optimize our current facilities for users through better ties to the networks and selective lines to tie individual users into more advantageous access points. We will also continue to explore other network and communication alternatives as they become available over the next year. Specific goals include improved response times and increased output speeds. TYMNET will be starting 1200 baud service soon and we would like to make it possible for users to take advantage of the higher output speed.

MAINSAIL:

We are awaiting the funding of the MAINSAIL project to allow initial export of this language system. We have established contacts with numerous outside groups interested in this machine-independent language ranging from university research projects to industry. (The university's research projects office is handling any problems or opportunities that may arise from proprietary values of these products, in accordance with established procedures). We have proposed an initial list of target

machines including PDP-10, PDP-11, and Nova. We plan to develop an exportable, documented form of the system for each of these environments and to test them in conjunction with appropriate collaborating user sites.

Adaptive User Interfaces:

We plan to continue work toward a more adaptive system for users including both simplifying access for non-expert users and anticipating default parameter conventions of individual users. We are now in the process of defining system calls which will make user specifications accessible to utility programs in a uniform way.

Software Facilities and Libraries:

There is a continuing need for improved documentation and self-learning facilities for various aspects of the system and of available programs. We will be up-grading this material, particularly as it relates to the inexperienced user.

We will continue to up-grade the various DEC-originated subsystems to the newest versions to increase the chance of compatibility. We have recently done this with FORTRAN and MACRO and will bring the other programs along as soon as possible. The whole issue of compatibility is one which will receive continued attention. We will also increase our mutual ties in software sharing with the TENEX and AI communities.

Some requests to look into additional software subsystems have been received and we will consider mounting them if the community develops a definite need.

Informal Information Access:

One characteristic of the SUMEX community is the diversity of information, formal and informal, which flows around the system or is available from users. We will continue to work on the HELP and BULLETIN BOARD systems to capture that information and direct it to other interested individuals. We will be working on capabilities both to ease the entry and cataloging of information and to assist in guiding the user to that subset which is of interest to him at a given time. These user-oriented lookup protocols are, of course, strongly related to the problems of adaptive user interfaces to the system and each will benefit from the experience of the other.

Community Management:

We will continue to work with the management committees to recruit the additional high quality projects which can be accommodated and to evolve resource allocation policies which appropriately reflect assigned priorities and project needs. We hope to make more generally available information about the various projects both inside and outside of the

community and thereby to promote the kinds of exchanges exemplified earlier and made possible by network facilities. The AIM workshops provide much useful information about the strengths and weaknesses of the performance programs both in terms of criticisms from other AI projects and in terms of the needs off practicing medical people. We plan to use this experience to guide the community building aspects of SUMEX-AIM.